06-12-00

A

PATENT

Attorney's. Docket No. 00-8010

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Box Patent Application
Assistant Commissioner for Patents
Washington, D. C. 20231

NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of:

Inventor(s): Stuart J. Jacobs
Francis L. Mannix, Jr.
Thomas W. Christoffel
Scott A. Belgard

For: **METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE**

Certification Under 37 CFR 1.10

I hereby certify that this New Application Transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service on this date 6/9/00 in an envelope as "Express Mail Post Office to Addressee" mailing label EK673491681US addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231.

Mary E. Anza
(typed or printed name of person mailing paper)

Mary E. Anza
(Signature of person mailing paper)

Attorney's Docket No. 00-8010

Enclosed are:

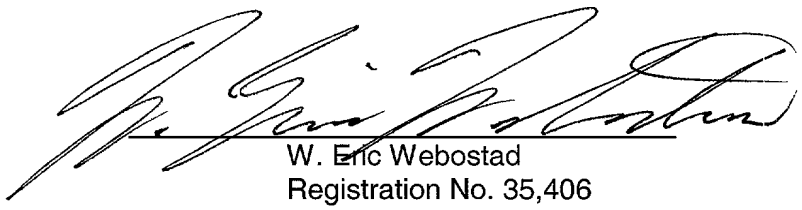
[15]	pages of specification and cover sheet
[7]	pages of claims
[1]	page of abstract
[5]	sheets of formal drawings.
[3]	pages of declaration and power of attorney.
[ ]	pages of assignment and assignment recordation form
[ ]	pages of information disclosure statement
[ ]	pages of form 1449
[ ]	references
[1]	return postcard

CLAIMS AS FILED

	NUMBER FILED	NUMBER EXTRA	RATE	BASIC FEE $690.00
TOTAL CLAIMS	23 - 20	3 x	$18.00	54.00
INDEPENDENT CLAIMS	7 - 3	4 x	$78.00	312.00
MULTIPLE DEPENDENT CLAIMS(S)		0 x	$260.00	0.00
			TOTAL FILING FEE	$1056.00

[x] Please charge my Deposit Account No. 07-2339 in the amount of **$1056.00.** **This transmittal letter is submitted in duplicate.**

[x] The Commissioner is hereby authorized to charge any additional fees under 37 CFR 1.16 and 1.17 which may be required by the papers submitted herewith or credit any overpayment to Account No. 07-2339.
This transmittal letter is submitted in duplicate.



W. Eric Webostad
Registration No. 35,406
Attorney for Applicant

GTE Service Corporation
600 Hidden Ridge, HQE03G13
Irving, TX 75038
Phone: (781) 466-4013
Fax: (781) 466-4021

EXPRESS MAIL NO. EK673491681US Docket No. 00-8010

UNITED STATES PATENT APPLICATION

OF

Stuart J. JACOBS

Francis L. MANNIX, Jr.

Thomas W. CHRISTOFFEL and

Scott A. BELGARD

FOR

METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE

METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE

GOVERNMENT CONTRACT

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Contract No. DAAL01-96-2-002 awarded by the U.S. Army.

FIELD OF THE INVENTION

The present invention relates generally to cryptography and, more particularly, to systems and methods for supporting cryptographic-related activities in a public key infrastructure.

BACKGROUND OF THE INVENTION

Public key cryptography has been commonly used to provide a mechanism to support access control and general authentication services in distributed environments. For example, in networks storing confidential information, conventional systems typically employ access control to limit access to the confidential information to designated parties. These systems may additionally employ general authentication services to authenticate users of various network resources to ensure that originators and recipients of messages are actually the parties they claim to be.

Conventional public key cryptography relies upon public key certificates, such as those defined in ITU X.509, to bind a user's public key reliably to his name and

provide users with the high level of assurance desired when identifying other entities. A certificate may be signed using a private key associated with the sender. The recipient of the message can then verify that the message was actually sent by the originator named in the message, provided that the recipient verifies the signature using the sender's public key.

In conventional systems, the responsibility for generating digital signatures and verifying digital signatures is borne by an application program executing on a general-purpose computer, under the control of a general-purpose operating system. For example, a conventional application program running on a computer in a network may have to authenticate link and network control messages. Similarly, the application program may have to encrypt messages using various encryption algorithms before transmitting these messages to other nodes in the network. Such tasks require an application programmer or network developer to acquire detailed knowledge of complex secret and public key algorithms and then develop programs to perform the required cryptographic functions.

Additionally, these cryptographic-related functions may require a network entity to exchange a number of messages with corresponding network entities when establishing a security association (SA). Such exchanges of messages consume very large amounts of network bandwidth, which is often limited in wireless systems.

As a result, there exists a need for a mechanism designed to avoid the requirement for an applications or network developer from having to acquire detailed knowledge of secret and public key algorithms. There is also a need for a mechanism

that integrates cryptographic-related functions into a simple to use service set, thereby simplifying the developer's task regarding inclusion of strong security features in application and infrastructure programs.

SUMMARY OF THE INVENTION

Systems and methods consistent with the present invention address these and other needs by integrating cryptographic-related functions in a software-based tool. The software-based tool includes a standardized interface that may be used by program developers for requesting the desired functions. The tool may also be customized to include any cryptographic-related functions based on the requirements of the particular user/system.

In accordance with the purpose of the invention as embodied and broadly described herein, a method for performing cryptographic-related functions in a network node is provided. The method includes receiving an input requiring cryptographic-related processing and generating a message based on the input. The message represents one of a predefined set of messages for processing by a cryptographic processing component. The method also includes transmitting the message to the cryptographic processing component and performing the cryptographic-related processing.

In another aspect of the present invention, a computer-readable medium, having sequences of instructions stored thereon is provided. The instructions may be invoked by a plurality of predefined messages and include sequences of instructions

which, when executed by a processor, cause the processor to receive an input representing one of the predefined messages. The instructions also cause the processor to transmit, based on the input, a request for cryptographic-related processing to a cryptographic processing module. The instructions further cause the processor to perform the cryptographic-related processing.

In still another aspect of the present invention, a cryptographic module is provided. The cryptographic module includes a memory configured to store a plurality of cryptographic processing programs where each program is invoked via one of a plurality of predefined messages. The cryptographic module also includes a processor configured to receive an input requiring cryptographic-related processing, generate one of the predefined messages based on the input, and transmit the message to a first one of the cryptographic processing programs. The processor is also configured to perform the cryptographic-related processing.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the principles of the invention. In the drawings,

Fig. 1 is a block diagram of an exemplary system in which an implementation consistent with the present invention may be employed;

Fig. 2 is an exemplary block diagram of a node of Fig. 1 in which systems and methods consistent with the present invention may be employed;

Fig. 3 illustrates user space components and kernel space components consistent with an implementation of the present invention;

Fig. 4 is an exemplary diagram of kernel space components consistent with an implementation of the present invention; and

Fig. 5 is a flowchart of processing for providing cryptographic-related functions in a manner consistent with the present invention.

DETAILED DESCRIPTION

The following detailed description of the invention refers to the accompanying drawings. The same reference numbers in different drawings identify the same or similar elements. Also, the following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims.

Systems and methods consistent with the present invention provide cryptographic-related functions in a software-based tool that may be used by an applications or network developer. The developer requests a particular function using a predefined set of messages transmitted to the software-based tool via a function call. The software-based tool then performs the desired cryptographic-related function.

EXEMPLARY SYSTEM CONFIGURATION

Fig. 1 is a diagram of an exemplary system 100 in which implementations of the present invention may be employed. The system 100 includes nodes 110, 120 and 130, server 140 and network 150.

Each of the nodes 110, 120 and 130 may include any type of computer device, such as a personal computer, a laptop, a personal digital assistant (PDA) or a similar device, with a connection to network 150. In an exemplary implementation of the present invention, the nodes 110-130 transmit/receive messages to/from other nodes over network 150 via wired, wireless, or optical connections. The network 150 may include the Internet, a local area network (LAN), wide area network (WAN), intranet or another type of network. Only three nodes are shown for simplicity. It should be understood, however, that any number of nodes may be included in system 100.

The server 140 may store certificates, public key information or other information required to verify/encrypt messages. For example, the server 140 may be a conventional light-weight directory access protocol (LDAP) server, an X.500 server or another type of server that stores certificates, certificate revocation lists (CRLs), or similar information. The nodes 110-130 may access the server 140 to retrieve various information needed to perform the authentication/verification functions, as described in more detail below.

EXEMPLARY NODE

Fig. 2 illustrates an exemplary node 110 of Fig. 1 in which methods and systems consistent with the present invention may be implemented. Node 110 includes a bus 210, a processor 220, a main memory 230, a read only memory (ROM) 240, a storage device 250, an input device 260, an output device 270, and a

communication interface 280. The bus 210 permits communication among the components of the node 110.

The processor 220 may include any type of conventional processor or microprocessor that interprets and executes instructions. Main memory 230 may be a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by processor 220. Main memory 230 may also store temporary variables or other intermediate information used during execution of instructions by processor 220. The ROM 240 may include a conventional ROM device or another type of static storage device that stores static information and instructions for processor 220. The storage device 250 may include any type of magnetic or optical recording medium and its corresponding drive, such as a magnetic disk or optical disk and its corresponding disk drive.

The input device 260 may include any conventional mechanism that permits an operator to input information to the node 110, such a keyboard, a mouse, a pen, voice recognition and/or biometric mechanisms, etc. The output device 270 may include any conventional mechanism that outputs information to the operator, including a display, a printer, a pair of speakers, etc. The communication interface 280 may include any transceiver-like mechanism that enables the node 110 to communicate with other devices and/or systems. For example, the communication interface 280 may include a modem or an Ethernet interface for communicating via a LAN. Alternatively, the communication interface 280 may include other mechanisms for communicating via a network, such as network 150.

Node 110, consistent with the present invention, performs cryptographic-related functions in response to processor 220 executing sequences of instructions contained in a computer readable medium, such as memory 230. A computer-readable medium may include one or more memory devices and/or carrier waves. Such instructions may be read into memory 230 from another computer-readable medium, such as a data storage device 250, or from a separate device via communication interface 280. Execution of the sequences of instructions contained in memory 230 causes processor 220 to perform the process steps that will be described hereafter. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

EXEMPLARY NODE COMPONENTS

Fig. 3 schematically illustrates the user space components and kernel space components in node 110, in an exemplary implementation consistent with the present invention. In the exemplary implementation, the user space components may reside in any one of or a combination of main memory 230, ROM 240 and storage device 250. In addition, the kernel space components may reside in any one of or a combination of main memory 230, ROM 240 and storage device 250. Additionally, both the user space components and kernel space components may interact with the other devices in node 110, such as processor 220.

The user space components and kernel space components are shown separated by a dotted line in Fig. 3. The user space components include user application program 310, public key authentication infrastructure (PKAI) control daemon 320, certificate database 330, PKAI operations daemon 340 and PKAI remote server daemon 350. The kernel space components include PKAI socket handler 360, PKAI call handler 370 and PKAI request handler 380.

The PKAI control daemon 320 initializes and shuts down PKAI services. A startup script may invoke the PKAI control daemon 320 with "start" and an optional pass phrase. The PKAI control daemon 320 may initialize the PKAI operations daemon 340 and the PKAI remote server daemon 350. In addition, the PKAI control daemon 320 shuts down the PKAI operations and remote server daemons 340 and 350 when the PKAI shuts down, such as when power to the node 110 is terminated.

The PKAI operations daemon 340 may communicate with PKAI request handler 380 via a user datagram protocol (UDP) socket. The PKAI operations daemon 340 performs local disk input/output on behalf of PKAI request handler 380. For example, the PKAI request handler 380 may use PKAI operations daemon 340 to store X.509 certificates to memory and retrieve X.509 digital certificates from memory, such as certificates database 330.

The PKAI remote server daemon 350 may also communicate with PKAI request handler via a UDP socket. The PKAI remote server daemon 350 retrieves information on behalf of PKAI request handler 380. For example, the PKAI remote server daemon 350 may retrieve X.509 digital certificates and certificate revocation

lists (CRLs) from a network accessible server, such as server 140 (Fig. 1). The PKAI remote server daemon 350 may store these certificates and CRLs in certificate database 330.

The PKAI system socket handler 360 may communicate with user application program 310 over a UDP socket. For example, the communication may include a request for cryptographic-related services, as discussed in more detail below. In this situation, the PKAI socket handler 360 generates a corresponding function call to the PKAI request handler 380 to perform the desired function. The PKAI call handler 370 may also receive system service calls from user application program 310 and PKAI control daemon 320. The PKAI call handler 370 then generates a corresponding function call to the PKAI request handler 380, based on the particular request.

Fig. 4 illustrates the PKAI components of Fig. 3 residing in the kernel space along with three cryptographic processing components, consistent with an exemplary implementation of the present invention. In an exemplary implementation, the cryptographic processing components are compiled into the kernel during a kernel re-build and their functionality invoked via a system service function call.

The kernel components illustrated in Fig. 4 include PKAI socket handler 360, PKAI call handler 370, PKAI request handler 380, PKAI RSA cryptoprocessing module 410, PKAI elliptic curve (EC) cryptoprocessing module 420 and PKAI keyed message digest algorithm 5 (MD5) cryptoprocessing module 430. Only three cryptoprocessing modules are shown for simplicity. Other cryptoprocessing modules

may be included in the kernel space based on the particular user/system requirements. Additionally, the details of the particular cryptoprocessing modules used in implementations of the present invention, such as modules 410-430, would be obvious to one of ordinary skill in this art and are not described herein.

Referring back to Fig. 3, a system service function call may be initiated by user application program 310 via the UDP socket to PKAI socket handler 360 or by a system service call directly to the PKAI call handler 370. The system service function call may also be initiated by PKAI control daemon 320 to PKAI call handler 370. Other methods of invoking the PKAI functions may also be used in alternative implementations. For example, the PKAI cryptographic-related functions may be invoked by any number of conventional call mechanisms based on the particular user/system requirements. In each case, the network or applications programmer need only be aware of the particular set of predefined messages needed to invoke the desired function. These messages are then incorporated into the user application program 310 as required.

As described above, the PKAI request handler 380 may receive requests for cryptographic-related services from a number of sources. In each situation, the PKAI request handler 380 receives the request and generates a function call to the appropriate cryptoprocessing module, such as one of cryptoprocessing modules 410-430. The details of performing the particular cryptographic-related functions are described in more detail below.

EXEMPLARY PROCESSING FOR PROVIDING CRYTPOGRAPHIC-RELATED FUNCTIONS

Fig. 5 illustrates processing associated with performing cryptographic-related functions using the PKAI system. Processing begins with initialization of the PKAI system (step 510). The PKAI system may be initialized by a startup script that is executed after node 110 is powered up. After the PKAI system begins operating, the control daemon, operations daemon and remote server daemons 320, 340 and 350, respectively, operate as described with regard to Figs. 3 and 4.

Assume that the user application program 310 executes an instruction requiring cryptographic-related processing (step 520). Such an instruction may, for example, require verifying a digital signature transmitted with a certificate using an RSA, MD5, EC or digital signature standard (DSS) algorithm or generating an RSA, MD5, EC or DSS digital signature. The instruction may also require encrypting or decrypting data using an RSA, EC or other cryptographic algorithm. The instruction may further require retrieving a digital certificate or certificate revocation list from either the user space components or a remote server, such as server 140. The instruction may also include verifying a certificate's hierarchy, performing self-signed certificate processing, performing certificate age checking, or retrieving, verifying and storing a digital certificate in the node. In essence, the instruction may require performing any cryptographic-related function, based on the system requirements.

The user application program 310, after executing the instruction, generates a system service call to PKAI call handler 370 (step 530). The system service call,

consistent with the present invention, may be chosen from a predefined list of messages that are used to invoke PKAI services. For example, assume that the request is for verifying an RSA 512 bit digital signature transmitted with a certificate. In this case, the predefined message may be PKAI_RSA512ver. In this scenario, the user application program 310 may then transmit PKAI_RSA512ver to the PKAI call handler 370. The PKAI call handler 370 receives the request and forwards a corresponding function call to the PKAI request handler 380 (step 540).

In the example, the PKAI call handler 370 may transmit PKAI_RSA52ver_req to the PKAI request handler 380. The PKAI request handler 380 may then transmit the request message to the appropriate cryptoprocessing module for processing (step 540). In this example, the PKAI request handler 380 transmits PKA_RSA512ver_req to PKAI RSA cryptoprocessing module 410 (Fig. 4). The PKAI RSA cryptoprocessing module 410 then performs the desired function, i.e., verifies the status of the RSA 512 bit digital signature transmitted with the certificate (step 550). The PKAI RSA cryptoprocessing module 410 then transmits the result to the PKAI request handler 380 (step 550). After receiving the result, the PKAI request handler 380 forwards the result back to the user application program 310 that initiated the request (step 560). The result may optionally be transmitted to the user application program 310 via the PKAI call handler 370.

Systems and methods consistent with the present invention enable applications and network programmers to incorporate any required cryptographic-related processing by merely incorporating the desired call message. An advantage of the

invention is that the applications or network programmer is able to incorporate complex security features without having to gain detailed knowledge of complex secret and public key algorithms. Appendix A illustrates an exemplary set of PKAI function call messages that may be used in an implementation consistent with the present invention. It should be understood that additional function call messages may be used in alternative implementations consistent with the present invention.

The foregoing description of preferred embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. For example, the PKAI system has been described as being resident in one of the network nodes that receives/transmits messages. In alternative implementations, the PKAI system may be located remotely from the network node. The scope of the invention is defined by the claims and their equivalents.

WHAT IS CLAIMED IS:

1. In a network node, a method for performing cryptographic-related functions, comprising:

receiving an input requiring cryptographic-related processing;

generating a message based on the input, the message representing one of a predefined set of messages for processing by a cryptographic processing component;

transmitting the message to the cryptographic processing component; and

performing the cryptographic-related processing.

2. The method of claim 1, wherein the cryptographic-related processing includes at least one of:

verifying or generating a digital signature; encrypting data; decrypting data; retrieving a digital certificate or certificate revocation list; verifying a certificate's hierarchy; self-signed certificate processing; retrieving, verifying and storing a digital certificate in the node; and certificate age checking.

3. The method of claim 1, wherein the node includes at least one application program, the method further comprising:

executing the input via the application program; and

wherein the generating a message includes:

generating the message via the application program.

4. The method of claim 1, wherein the node includes at least one application program, the method further comprising:

generating an output message via the application program, the output message requiring cryptographic-related processing;

transmitting, based on the required cryptographic-related processing, one of the predefined set of messages to the cryptographic processing component;

performing the cryptographic-related processing; and

outputting the processed message.

5. A computer-readable medium having stored thereon a plurality of sequences of instructions that may be invoked by a plurality of predefined messages, said instructions including sequences of instructions which, when executed by a processor, cause said processor to perform the steps of:

receiving an input representing one of the predefined messages;

transmitting, based on the input, a request for cryptographic-related processing to a cryptographic processing module; and

performing the cryptographic-related processing.

6. The computer-readable medium of claim 5, wherein the performing the cryptographic-related processing includes at least one of:

verifying or generating a digital signature; encrypting or decrypting data; retrieving a digital certificate or certificate revocation list; verifying a certificate's

hierarchy; self-signed certificate processing; retrieving, verifying and storing a digital certificate; and certificate age checking.

7. The computer-readable medium of claim 5, wherein the transmitting includes:

sending a function call to the cryptographic processing module.

8. The computer-readable medium of claim 5, wherein the input represents a digitally signed network control message requiring verification.

9. A cryptographic module, comprising:

a memory configured to store a plurality of cryptographic processing programs, each program being invoked via one of a plurality of predefined messages; and

a processor configured to:

receive an input requiring cryptographic-related processing,

generate one of the predefined messages based on the input,

transmit the message to a first one of the cryptographic processing programs, and

perform the cryptographic-related processing.

10. The cryptographic module of claim 9, wherein when performing the cryptographic-related processing, the processor is configured to perform at least one of:

verifying or generating a digital signature; encrypting data; decrypting data; retrieving a digital certificate or certificate revocation list; verifying a certificate's hierarchy; self-signed certificate processing; retrieving, verifying and storing a digital certificate; and certificate age checking.

11. The cryptographic module of claim 9, wherein when transmitting the message, the processor is further configured to:

transmit a function call to the first cryptographic processing program.

12. The cryptographic module of claim 9, wherein the processor is further configured to:

transmit the result of the cryptographic-related processing to an application program.

13. A cryptographic module, comprising:

means for storing a plurality of cryptographic processing programs, each program being invoked via one of a plurality of predefined messages;

means for receiving an input requiring cryptographic-related processing;

means for generating one of the predefined messages based on the input;

means for transmitting the message to a first one of the cryptographic processing programs; and

means for performing the cryptographic-related processing.

14. In a node coupled to other nodes in a network, the node including an application program for handling communications with the other nodes, a method of performing cryptographic-related functions, the method comprising:

receiving an input requiring a cryptographic-related operation;

generating a predefined message based on the input, the message representing one of a plurality of predefined messages usable by a cryptographic processing program;

transmitting the predefined message to the cryptographic processing program; and

performing, via the cryptographic processing program, the desired cryptographic-related operation.

15. The method of claim 14, further comprising:

returning the result of the performing to the application program.

16. The method of claim 14, wherein the predefined message includes at least one of:

a request for digital signature generation, a request for digital signature verification, a request for data encryption, a request for data decryption, a request for retrieval of a digital certificate, a request for retrieval of a certificate revocation list, a request for verification of a certificate's hierarchy, a request for self-signed certificate processing, and a request for certificate age checking.

17. The method of claim 16, wherein the request for digital signature generation includes a request for at least one of RSA signature generation, secret keyed MD5 signature generation, elliptic curve signature generation and digital signature standard signature generation.

18. The method of claim 16, wherein the request for digital signature verification includes a request for at least one of RSA signature verification, secret keyed MD5 signature verification, elliptic curve signature verification and digital signature standard signature verification.

19. The method of claim 16, wherein the request for data encryption includes a request for at least one of RSA based encryption and elliptic curve based encryption.

20. The method of claim 16, wherein the request for data decryption includes a request for at least one of RSA based decryption and elliptic curve based decryption.

21. The method of claim 14, wherein the performing includes:

accessing a remote server via the network to retrieve cryptographic-related information.

22. A computer-readable medium that stores instructions executable by at least one processor to perform a method for providing cryptographic-related functions, comprising:

receiving a first function call from a predefined list of function calls, the predefined list of function calls representing available cryptographic-related functions executable by the at least one processor;

generating a request message based on the first function call, the request message representing a request for processing by a cryptographic processing module;

transmitting the request message to the cryptographic processing module; and

performing the cryptographic-related function.

23. A system for performing cryptographic-related functions, comprising:

a call handler component configured to receive a function call from an application program and generate a request message;

a request handler configured to receive the request message and generate a corresponding instruction request; and

a cryptographic processing component configured to receive the instruction request and perform cryptographic-related processing.

ABSTRACT

In a node (110) communicating with other nodes in a network (150), a system and method for performing cryptographic-related functions is provided. The node (110) receives and transmits inputs and outputs requiring cryptographic-related processing. When cryptographic processing is required, the node (110) transmits a predefined message to a cryptographic processing component in the node (110) that then performs the desired cryptographic-related processing.

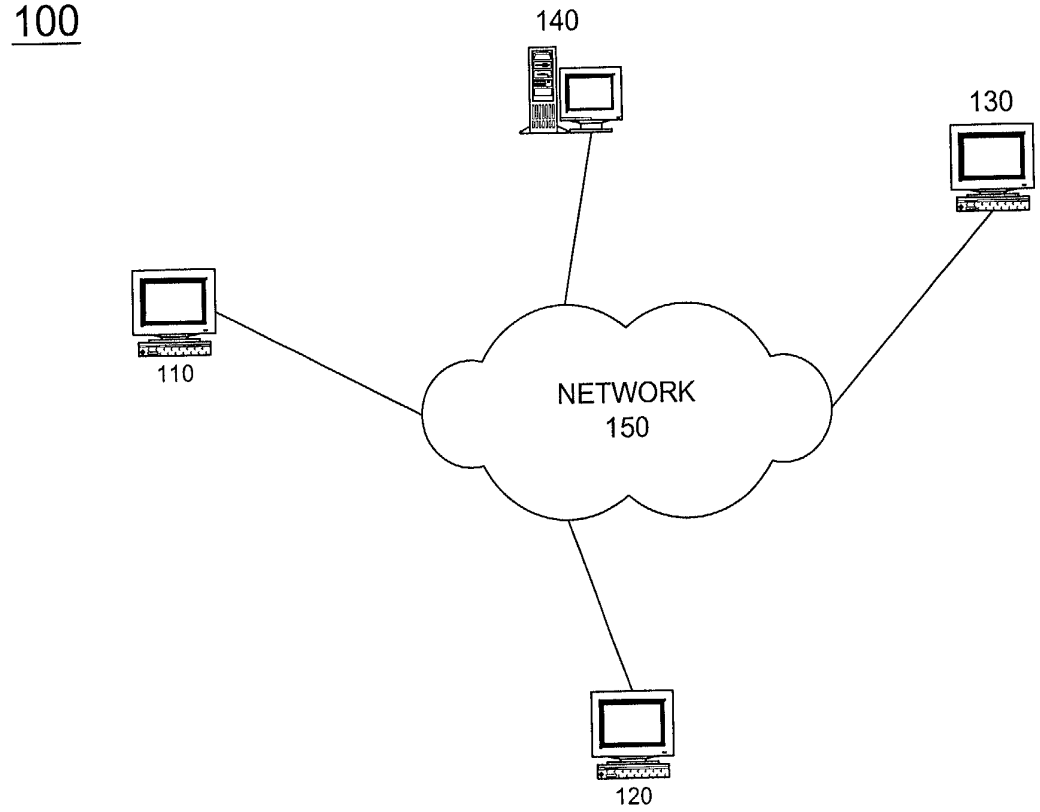


FIG. 1

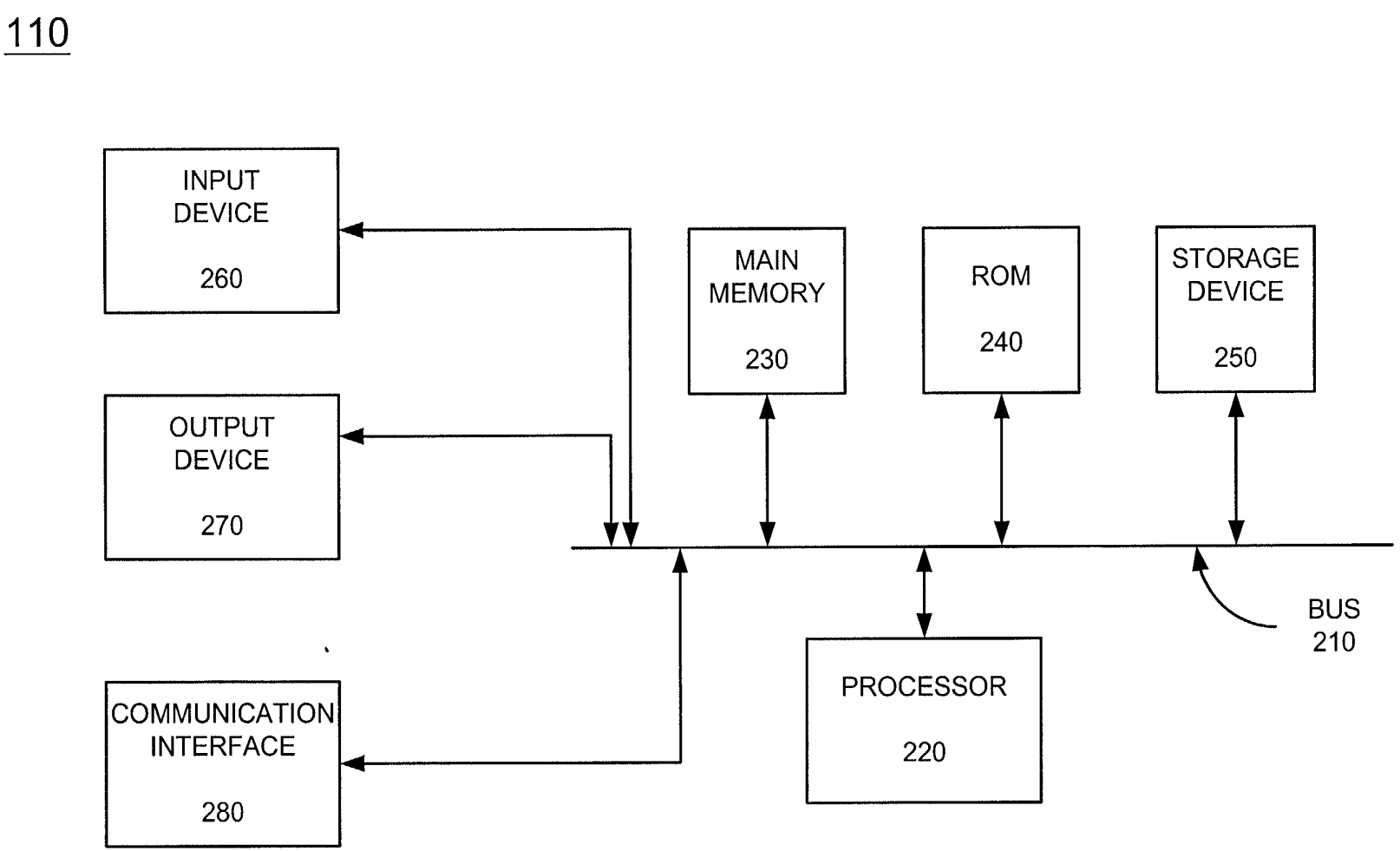


FIG. 2

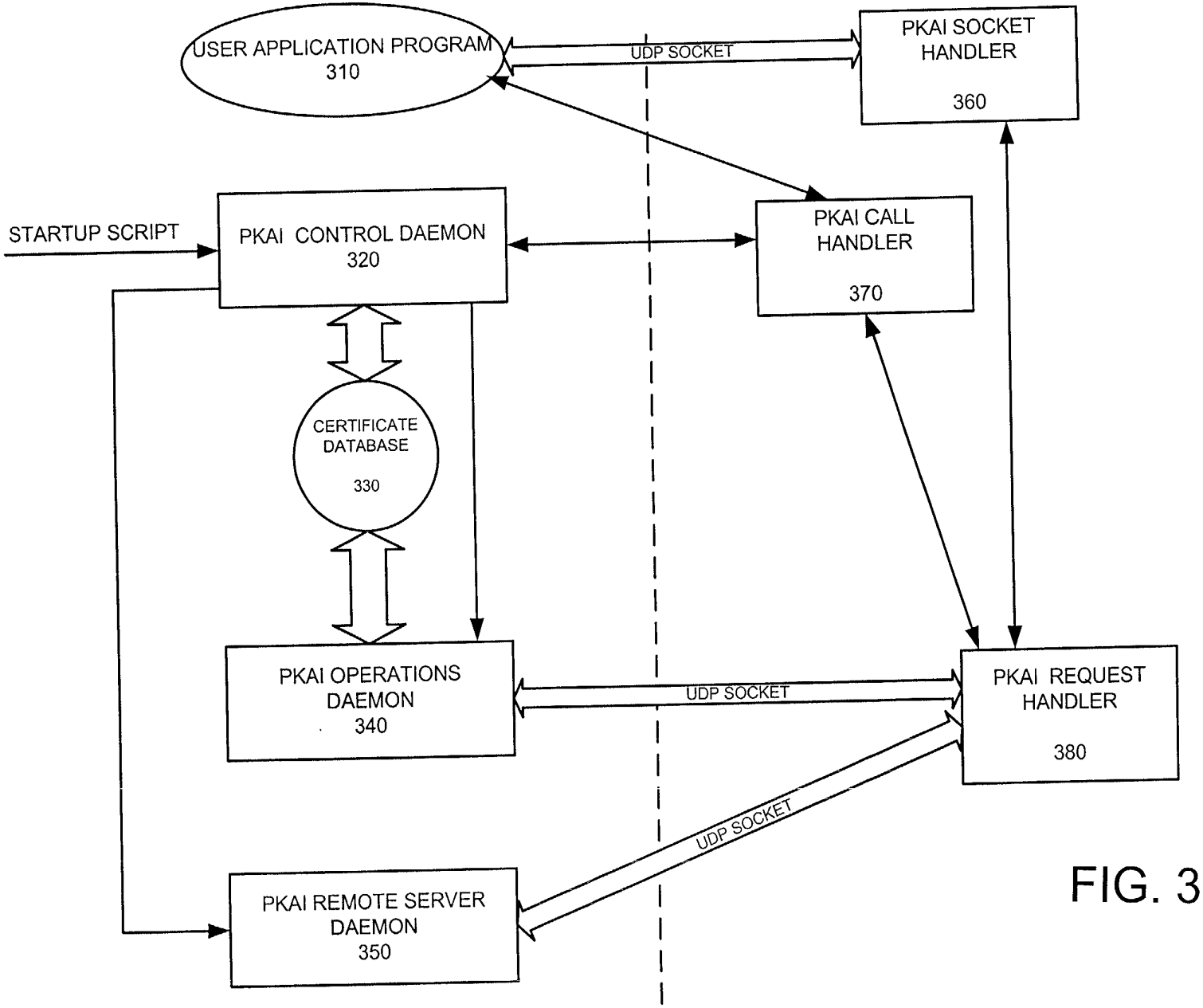


FIG. 3

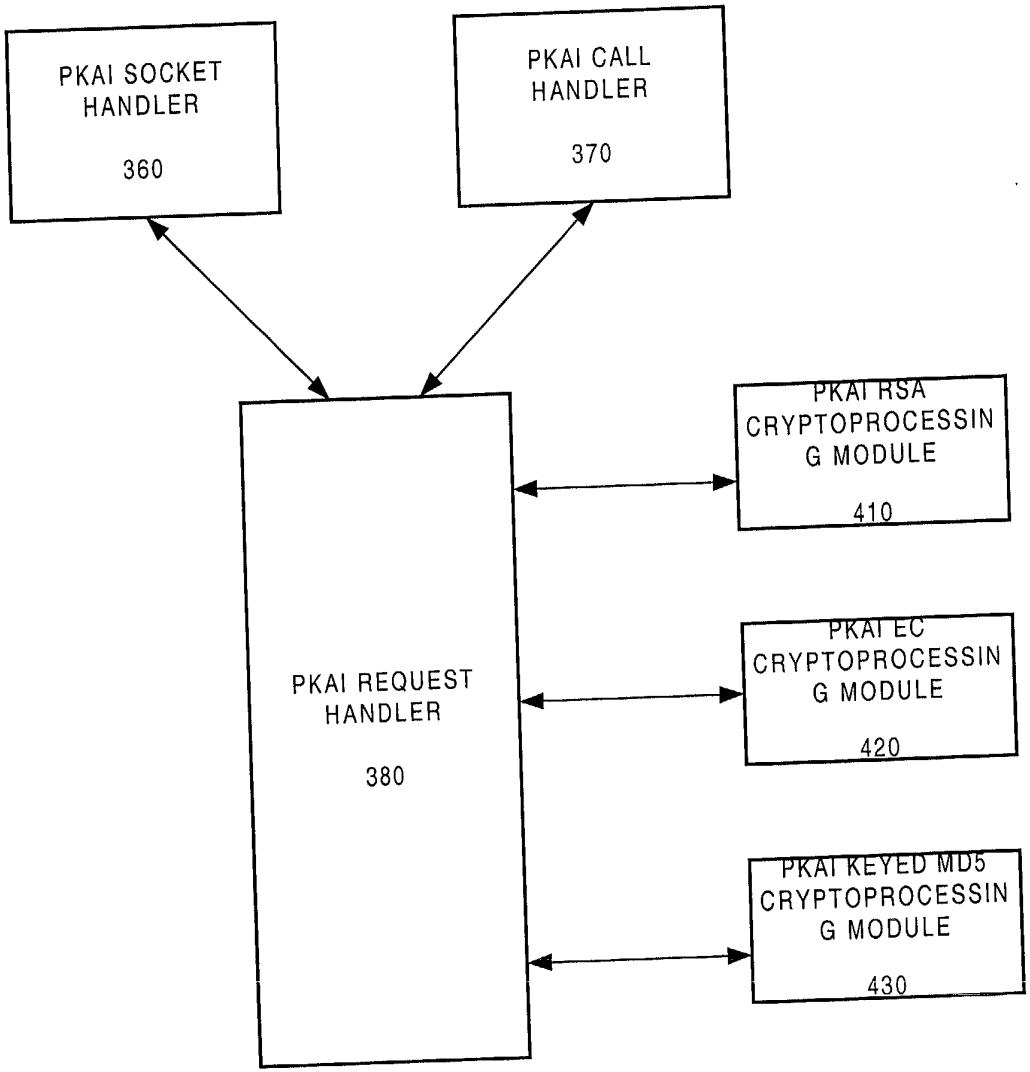


FIG. 4

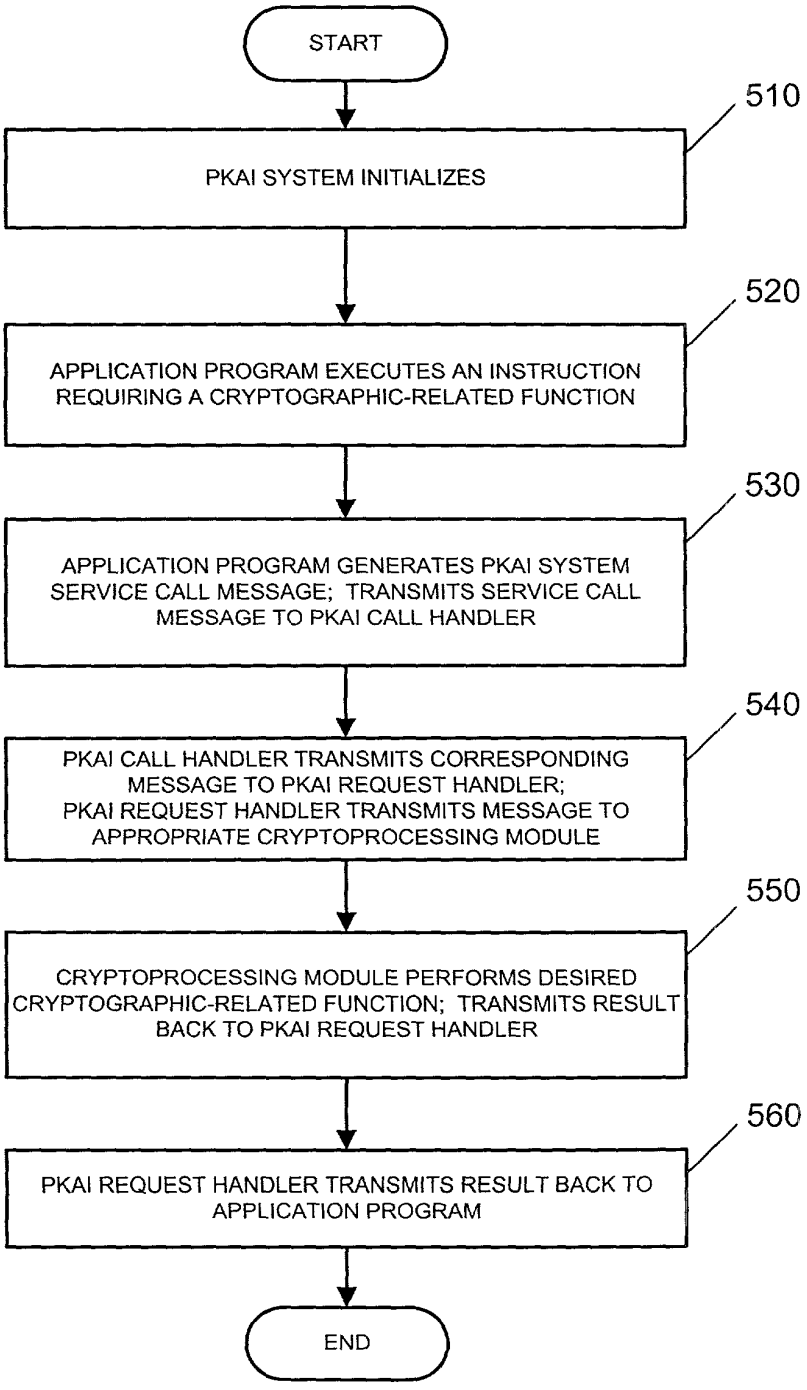

START
510
PKAI SYSTEM INITIALIZES
520
APPLICATION PROGRAM EXECUTES AN INSTRUCTION REQUIRING A CRYPTOGRAPHIC-RELATED FUNCTION
530
APPLICATION PROGRAM GENERATES PKAI SYSTEM SERVICE CALL MESSAGE; TRANSMITS SERVICE CALL MESSAGE TO PKAI CALL HANDLER
540
PKAI CALL HANDLER TRANSMITS CORRESPONDING MESSAGE TO PKAI REQUEST HANDLER;
PKAI REQUEST HANDLER TRANSMITS MESSAGE TO APPROPRIATE CRYPTOPROCESSING MODULE
550
CRYPTOPROCESSING MODULE PERFORMS DESIRED CRYPTOGRAPHIC-RELATED FUNCTION; TRANSMITS RESULT BACK TO PKAI REQUEST HANDLER
560
PKAI REQUEST HANDLER TRANSMITS RESULT BACK TO APPLICATION PROGRAM
END

FIG. 5

EXPRESS MAIL NO. EK673491681US

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Attorney Docket No. **00-8010**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR SUPPORTING CRYPTOGRAPHIC-RELATED ACTIVITIES IN A PUBLIC KEY INFRASTRUCTURE

the specification of which (check one) [x] is attached hereto. [ ] was filed on as Appln. Serial No. and was amended on (if applicable). I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
(Number)	(Country)	(Day/Month/Year filed)	[_] Yes [_] No

I hereby claim the benefit under Title 35, United States Code, 119(e) of any United States provisional applications(s) listed below.

(Application Number) (Filing Date)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date for this application:

(Appln. Serial No.) (Filing Date) (Status--patented, pending, abandoned)

Attorney Docket No. **00-8010**

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Leonard C. Suchyta, Reg. No. 25,707, Floyd E. Anderson, Reg. No. 33,825 and W. Eric Webostad, Reg. No. 35,406

Address all telephone calls to **W. Eric Webostad** at telephone no. **(781) 466-4013**

Address all correspondence to **Leonard C. Suchyta**
GTE Service Corporation
600 Hidden Ridge, HQE03G13
Irving, TX 75038

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF SOLE OR FIRST INVENTOR **Stuart J. Jacobs**

Inventor's signature _____ Date _____

Residence **Hudson, Massachusetts** Citizenship **USA**

Post Office Address **10 Edith Rd., Hudson, MA 01749**

FULL NAME OF SECOND INVENTOR **Francis Leo Mannix, Jr.**

Inventor's signature _____ Date _____

Residence **Newton, Massachusetts** Citizenship **USA**

Post Office Address **16 George St., Newton, MA 02458**

FULL NAME OF THIRD INVENTOR **Thomas William Christoffel**

Inventor's signature _____ Date _____

Residence **Concord, Massachusetts** Citizenship **USA**

Post Office Address **256 Bedford St., Concord, MA 01742**

Attorney Docket No. **00-8010**

FULL NAME OF FOURTH INVENTOR **Scott Andrew Belgard**

Inventor's signature _____ Date _____

Residence **Randolph, Massachusetts** Citizenship **USA**

Post Office Address **31 Niles Rd., Randolph, MA 02368**

APPENDIX A

Each table provides the function name and a brief usage description.

Function Name	Function Usage
pkail_cert_check	Verify a certificate as having valid "not before" and "not after" dates, and that other certificate fields are correct.
pkail_cert_create	Create a certificate entry. calls pkail_cert_decompose, pkail_cert_sig_check, pkail_cert_check
pkail_cert_decompose	Decompose a certificate into it's individual fields and move these field values into pkail_cert_fields structure
pkail_cert_sig_check	Verify a certificate as having valid digital signature. Calls pkail_rsaverify
pkail_disk_getcert	Request pkail_oper daemon to retrieve a ASN.1 BER formatted cert from pkail disk located database
pkail_disk_putcert	Request pkail_oper daemon to store a ASN.1 BER formatted cert in pkail disk located database
pkail_fndcert	Locate and return ASN.1 BER formatted cert to requester. Calls pkail_disk_getcert
pkail_loadcert	Process new certificate received from user space for loading into pkail kernel storage and pkail disk database. Calls pkail_cert_create, pkail_disk_putcert

Table 1 PKAI Certificate Specific kernel Functions

Function Name	Function Usage
pkail_keyedmd5sign	Generate a prefix-postfix keyed MD5 digital signature. Calls pkail_skey_get, pkail_md5
pkail_keyedmd5ver	Verify a prefix-postfix keyed MD5 digital signature Calls pkail_skey_get, pkail_md5
pkail_md5	Generate secret Prefix Postfix keyed MD5 message hash operation resulting in a secret key digital signature
pkail_setspi	Process a secret key entry for secret keys used with the keyed MD5 authentication. Calls pkail_skey_get, pkail_skey_create
pkail_skey_create	Create a secret key entry for secret keysin pkail_skey_list
pkail_skey_get	Locate a secret key entry by searching the pkail_skey_list for a match on pkail_skey_list.addr and pkail_skey_list.spi and when found return pkail_skey_list.key of pkail_skey_list.klen

Table 2 PKAI Secret Key Specific kernel Functions

APPENDIX A

Function Name	Function Usage
pkail_rsasign	Process request to generate an RSA digital signature. Calls pkail_privkey_get, pkail_rsa512sign, pkail_rsa768sign, pkail_rsa1024sign, pkail_rsa2048sign
pkail_rsa512sign	Generate an RSA 512 bit digital signature
pkail_rsa768sign	Generate an RSA 768 bit digital signature
pkail_rsa1024sign	Generate an RSA 1024 bit digital signature
pkail_rsa2048sign	Generate an RSA 2048 bit digital signature
pkail_rsaverify	Verify an RSA digital signature. Calls pkail_pubkey_get, pkail_rsa512ver, pkail_rsa768ver, pkail_rsa1024ver, pkail_rsa2048ver
pkail_rsa512ver	Verify an RSA 512 bit digital signature
pkail_rsa768ver	Verify an RSA 7688 bit digital signature
pkail_rsa1024ver	Verify an RSA 1024 bit digital signature
pkail_rsa2048ver	Verify an RSA 2048 bit digital signature
pkail_rsaencryp	Encrypt up to 4096 bytes of requester text. Calls pkail_privkey_get, pkail_rsa512encryp, pkail_rsa768encryp, pkail_rsa1024encryp, pkail_rsa2048encryp
pkail_rsa512encryp	Encrypt up to 4096 bytes of requester text using an RSA 512 bit public key
pkail_rsa768encryp	Encrypt up to 4096 bytes of requester text using an RSA 768 bit public key
pkail_rsa1024encryp	Encrypt up to 4096 bytes of requester text using an RSA 1024 bit public key
pkail_rsa2048encryp	Encrypt up to 4096 bytes of requester text using an RSA 2048 bit public key
pkail_rsadecryp	Decrypt up to 4096 bytes of requester text. Calls pkail_pubkey_get, pkail_rsa512decryp, pkail_rsa768decryp, pkail_rsa1024decryp, pkail_rsa2048decryp
pkail_rsa512decryp	Decrypt up to 4096 bytes of requester text using an RSA 512 bit private key
pkail_rsa768decryp	Decrypt up to 4096 bytes of requester text using an RSA 768 bit private key
pkail_rsa1024decryp	Decrypt up to 4096 bytes of requester text using an RSA 1024 bit private key
pkail_rsa2048decryp	Decrypt up to 4096 bytes of requester text using an RSA 2048 bit private key

Table 3 PKAI RSA Specific kernel Functions

APPENDIX A

Function Name	Function Usage
pkail_ecsign	Process request to generate an EC digital signature. Calls pkail_privkey_get, pkail_ec80sign, pkail_ec120sign, pkail_ec1160sign
pkail_ec80sign	Generate an EC 80 bit digital signature
pkail_ec120sign	Generate an EC 120 bit digital signature
pkail_ec160sign	Generate an EC 160 bit digital signature
pkail_ecverify	Verify an EC digital signature. Calls pkail_pubkey_get, pkail_ec80ver, pkail_ec120ver, pkail_ec160ver
pkail_ec80ver	Verify an EC 80 bit digital signature
pkail_ec120ver	Verify an EC 120 bit digital signature
pkail_ec160ver	Verify an EC 160 bit digital signature
pkail_ecencryp	Encrypt up to 4096 bytes of requester text. Calls pkail_privkey_get, pkail_ec80encryp, pkail_ec120encryp, pkail_ec160encryp
pkail_ec80encryp	Encrypt up to 4096 bytes of requester text using an EC 80 bit public key
pkail_ec120encryp	Encrypt up to 4096 bytes of requester text using an EC 120 bit public key
pkail_ec160encryp	Encrypt up to 4096 bytes of requester text using an EC 160 bit public key
pkail_ecdecryp	Decrypt up to 4096 bytes of requester text. Calls pkail_pubkey_get, pkail_ec80decryp, pkail_ec120decryp, pkail_ec160decryp
pkail_ec80decryp	Decrypt up to 4096 bytes of requester text using an EC 80 bit private key
pkail_ec120decryp	Decrypt up to 4096 bytes of requester text using an EC 120 bit private key
pkail_ec160decryp	Decrypt up to 4096 bytes of requester text using an EC 160 bit private key

Table 4 PKAI EC Specific kernel Functions

Function Name	Function Usage
pkail_dsssign	Generate a DSS 512 bit digital signature
pkail_dssverify	Verify a DSS 512 bit digital signature

Table 5 PKAI DSS Specific kernel Functions

APPENDIX A

Function Name	Function Usage
pkail_cleartables	Clear out all pkail kernel tables by overwriting with zeros, de-allocating memory, close sockets from pkail kernel services to pkail daemons
pkail_core	Identify requested pkail kernel service and call required action functions
pkail_err	General pkail kernel services function for logging errors to sys log
pkail_opensockets	Open sockets from pkail kernel services to pkail daemons
pkail_pphrase	process part of passphrase received from user space
pkail_privkey_create	create a private key entry. Calls pkail_privkey_get
pkail_privkey_get	Locate a private key, search the pkail_privkey_list for a match on either pkail_privkey_list.my_ip_addr or pkail_privkey_list.my_host_name
pkail_pubkey_get	Locate a public key, search the pkail_cert_list for a match on either pkail_cert_list.subj_ip_addr or pkail_cert_list.subj_host_name and when found return pkail_cert_list.pubkey
sys_soc_pkail	Receive UDP socket service calls from user space clients of pkail kernel services.
sys_pkail	Receive system service calls from user space clients of pkail kernel services. Calls verify_area and pkail_core

Table 6 PKAI Common kernel Functions